Adaptive Single–Trial Error/Erasure Decoding for Binary Codes

Christian Senger, Vladimir R. Sidorenko, Steffen Schober, Martin Bossert
Inst. of Telecommunications and Applied Information Theory
Ulm University, Ulm, Germany
{christian.senger | vladimir.sidorenko | steffen.schober | martin.bossert}@uni-ulm.de

Victor V. Zyablov
Inst. for Information Transmission Problems
Russian Academy of Sciences, Moscow, Russia
zyablov@iitp.ru

***Abstract*—We investigate adaptive single–trial error/erasure decoding of binary codes whose decoder is able to correct ε errors and τ erasures if $\lambda\varepsilon + \tau \leq d_{\min} - 1$. Thereby, $d_{\min}$ is the minimum Hamming distance of the code and $1 < \lambda \leq 2$ is the tradeoff parameter between errors and erasures. The error/erasure decoder allows to exploit soft information by treating a set of most unreliable received symbols as erasures. The obvious question here is, how this erasing should be performed, i.e. how the unreliable symbols which must be erased to obtain the smallest possible residual codeword error probability are determined. This was answered before [1] for the case of fixed erasing, where only the channel state and not the individual symbol reliabilities are taken into consideration. In this paper, we address the adaptive case, where the optimal erasing strategy is determined for every given received vector.**

I. Introduction

The idea of exploiting soft information from the transmission channel using hard–decision algebraic error/erasure decoders dates back to Forney [2], [3]. His *Generalized Minimum Distance (GMD)* decoding scheme applies a *Bounded Minimum Distance (BMD)* error/erasure decoder repeatedly, each time with a different number of erased most unreliable received symbols. Forney proved that the residual codeword error probability of GMD decoding approaches that of *Maximum Likelihood (ML)* decoding if the channel is good and the number of decoding trials is $\frac{d_{\min}}{2}$, where $d_{\min}$ is the minimum *Hamming distance* of the code. This explains why GMD decoding is frequently applied for concatenated coding schemes. There, the inner code is responsible for correcting a considerable amount of transmission channel errors. Thus, the input symbols for the outer decoder can be viewed as being transmitted over a *super channel*, which is composed of the transmission channel and the inner decoder. This super channel is always good if the parameters of the inner code are chosen appropriately.

The fundamental task of GMD decoding with given number of decoding trials is to find an erasing strategy which either maximizes the guaranteed decoding radius or minimizes the residual codeword error probability. Both measures can be optimized either in a fixed manner or adaptively. For fixed erasing, the erasing strategy depends only on the state of the transmission channel and remains unchanged for each received vector. The fixed approach essentially optimizes the overall worst–case measure. Adaptive erasing on the other hand takes every single received vector into consideration, choosing the optimal erasing strategy for exactly this specific received vector. Obviously, one can expect the adaptive approach to yield better results than the fixed approach, especially for mediocre channel conditions.

Different settings for optimal fixed erasing have been considered in [4]–[6] (radius maximization), [1], [7] (error probability minimization), and [8], [9] (both). Results about adaptive erasing can be found in [6], [10]–[13] (radius maximization).

In the present paper, we tackle the previously unconsidered problem of adaptive erasing with the target of minimizing the residual codeword error probability. In doing so, we restrict ourselves to one single decoding trial. This restriction allows to focus on the core of the problem and will be relaxed in future work. Furthermore, we assume binary antipodal signaling and a memoryless channel with soft output. The *Additive White Gaussian Noise (AWGN)* channel will serve as our main example for such channels.

The paper is organized as follows. In Section II, we briefly describe error/erasure decoding and introduce some required notations. In Section III, we derive an adaptive erasing strategy which minimizes the residual codeword error probability. In doing so, we apply basic techniques from probability theory like discrete random variables and probability generating functions. A computationally more efficient version of the erasing strategy is given in Section IV. Simulation results are given in Section V, conclusions and an outlook to further research in Section VI.

II. Error/Erasure Decoding

We consider a binary code $\mathcal{C}(2; n, k, d_{\min})$ with length n, dimension k and minimum Hamming distance $d_{\min}$. For $\mathcal{C}$, we have a *λ–extended Bounded Distance* error/erasure decoder or simply *λ–decoder* $\mathrm{dec}_{\mathcal{C}}(\cdot)$ which is able to correct ε errors and τ erasures if $\lambda\,\varepsilon + \tau \leq d_{\min} - 1$. Here, $1 < \lambda \leq 2$ is the tradeoff parameter between errors and erasures. For $\lambda = 2$, the decoder is a traditional BMD error/erasure decoder. For *Bose–Chaudhuri–Hocquenghem (BCH)* codes, such error/erasure decoders are described e.g. in [14], [15].

This work has been supported by DFG, Germany, under grants BO 867/17, BO 867/21-1, and RUS 436/113/0. Vladimir Sidorenko is on leave from IITP, Russian Academy of Sciences, Moscow, Russia.

At the transmitter, an information vector $\mathbf{a} \in \mathbb{F}_2^k$ is encoded into a codeword $\mathbf{c} \in \mathcal{C} \subseteq \mathbb{F}_2^n$. The binary symbols c_i, $i = 0, \ldots, n-1$, are then mapped to binary antipodal signals $x_i := -1^{c_i} \in \{-1, +1\}$, which are transmitted over the channel. Each transmitted symbol x_i is distorted by the channel to a received symbol $y_i \in \mathbb{R}$. The λ–decoder can only handle hard input, hence the real received symbols must be mapped to symbols of the binary field $\mathbb{F}_2$. This can be accomplished by the *Heaviside–like function*

$$\alpha := \begin{cases} \mathbb{R} & \longrightarrow & \mathbb{R} \\ y & \longmapsto & \begin{cases} -1, & \text{if } y \leq 0 \\ +1, & \text{if } y > 0 \end{cases} \end{cases},$$

which essentially extracts the sign of a real received symbol, and the inverse mapping function

$$\beta := \begin{cases} \{-1, +1\} & \longrightarrow & \mathbb{F}_2 \\ y & \longmapsto & \begin{cases} 1, & \text{if } y = -1 \\ 0, & \text{if } y = +1 \end{cases} \end{cases},$$

which maps real symbols to symbols of $\mathbb{F}_2$. The binary vector

$$\mathbf{r} := (\beta(\alpha(y_0)), \ldots, \beta(\alpha(y_{n-1}))) \tag{1}$$

is a distorted version of the transmitted codeword $\mathbf{c}$ and could be fed into the λ–decoder for traditional errors–only decoding. Decoding would be successful if for the number $\varepsilon := \mathrm{d_H}(\mathbf{c}, \mathbf{r})$ of errors in $\mathbf{r}$ holds $\lambda\,\varepsilon \leq d_{\min} - 1$ or, in more familiar notation, $\varepsilon \leq \left\lfloor \frac{d_{\min}-1}{\lambda} \right\rfloor$. Here, $\mathrm{d_H}(\cdot, \cdot)$ is the Hamming distance between two vectors of equal length.

Let $\mathrm{P}_\sigma(\cdot|\cdot)$ be the transition probability of the memoryless channel, the parameter σ marks the channel state. Then, using *Bayes' Theorem*, we can calculate for each received symbol y the probability $h_\sigma(y)$ that $-\alpha(y)$ was transmitted, i.e. a transmission error occurred.

$$\begin{aligned} h_\sigma(y) &:= \mathrm{P}_\sigma(-\alpha(y)\,|\,y) \\ &= \frac{\mathrm{P}_\sigma(y\,|\,-\alpha(y))\,\Pr(-\alpha(y))}{\Pr(y)} \\ &= \frac{\mathrm{P}_\sigma(y\,|\,-\alpha(y))\,\Pr(-\alpha(y))}{\mathrm{P}_\sigma(y\,|\alpha(y))\,\Pr(\alpha(y)) + \mathrm{P}_\sigma(y\,|\,-\alpha(y))\,\Pr(-\alpha(y))} \\ &= \frac{\mathrm{P}_\sigma(y\,|\,-\alpha(y))}{\mathrm{P}_\sigma(y\,|\alpha(y)) + \mathrm{P}_\sigma(y\,|\,-\alpha(y))}, \end{aligned}$$

where the last equality follows from the reasonable assumption $\Pr(-\alpha(y)) = \Pr(\alpha(y)) = \frac{1}{2}$ of equiprobable transmitted symbols. It is justified to denote $h_\sigma(y)$ as *unreliability value* of the received symbol y. The greater $h_\sigma(y)$, the higher the probability that y is an erroneous symbol. W.l.o.g. let us from now on assume that the symbols of the received vector $\mathbf{y}$ (and by (1) also $\mathbf{r}$) are ordered according to their unreliability value, i.e. $h_\sigma(y_0) \geq \cdots \geq h_\sigma(y_{n-1})$.

We obtain a new received vector by erasing the τ most unreliable symbols in $\mathbf{r}$. This new vector is denoted by

$$\mathbf{r}_\tau := (\underbrace{\text{X}, \ldots, \text{X}}_{\tau \text{ times}}, r_\tau, \ldots, r_{n-1}).$$

The λ–decoder is capable of decoding $\mathbf{r}_\tau$ as long as $\lambda\,\varepsilon + \tau \leq d_{\min} - 1$, where ε is the number of errors in the non–erased symbols $r_\tau, \ldots, r_{n-1}$. The number of erasures is the decoder's degree of freedom, so the task of an adaptive error/erasure decoder is as follows.

Problem 1 *For given received vector* $\mathbf{y} = (y_0, \ldots, y_{n-1})$ *with ordered unreliabilities* $h_\sigma(y_0) \geq \cdots \geq h_\sigma(y_{n-1})$ *and channel state* σ *find the optimal number* $0 \leq \tau_\sigma^\star \leq d_{\min} - 1$ *of erased most unreliable symbols such that the residual codeword error probability of decoding* $\mathbf{r}_{\tau^\star}$ *with the* λ*–decoder* $\mathrm{dec}_\mathcal{C}(\cdot)$ *is minimized.*

In the following section we provide an exact solution to Problem 1 which is computationally expensive. In Section IV we give a very good approximated solution which is computationally efficient.

III. Derivation of an Adaptive Erasing Strategy

To solve Problem 1 it is required to express the residual codeword error probability after adaptive error/erasure decoding as a function of the number τ of erased symbols. We accomplish this using basic techniques from probability theory.

Let the discrete random variables X_i, $i = 0, \ldots, n-1$ be defined by

$$X_i := \begin{cases} 1, & \text{if } y_i \text{ is erroneous } (y_i \neq x_i) \\ 0, & \text{if } y_i \text{ is correct } (y_i = x_i) \end{cases}.$$

The probabilities of the two possible values of X_i are determined by the unreliability value of symbol y_i, i.e. $\Pr(X_i = 1) = h_\sigma(y_i)$ and $\Pr(X_i = 0) = 1 - h_\sigma(y_i)$.

Since X_i takes on only nonnegative integer values, its *probability generating function (PGF)* [16], [17] is given by

$$\begin{aligned} G_{\sigma,X_i}(\rho) &:= \mathrm{E}\{\rho^{X_i}\} \\ &= \Pr(X_i = 0) + \rho\,\Pr(X_i = 1) \\ &= 1 - h_\sigma(y_i) + \rho h_\sigma(y_i). \end{aligned} \tag{2}$$

Assume that the τ most unreliable symbols of $\mathbf{r}$ are erased and $\mathbf{r}_\tau$ is fed into the λ–decoder. Then, there are ε, $0 \leq \varepsilon \leq n - \tau$, erroneous symbols among the non–erased $n - \tau$ symbols. We can model their number with a new random variable Y_τ using the random variables X_i, $i = \tau, \ldots, n-1$.

$$Y_\tau := \sum_{i=\tau}^{n-1} X_i.$$

We obtain

$$\begin{aligned} G_{\sigma,Y_\tau}(\rho) &:= \mathrm{E}\{\rho^{Y_\tau}\} \\ &= \mathrm{E}\{\rho^{X_\tau + \cdots + X_{n-1}}\} \\ &= \mathrm{E}\{\rho^{X_\tau} \cdot \cdots \cdot \rho^{X_{n-1}}\} \qquad (3) \\ &= \mathrm{E}\{\rho^{X_\tau}\} \cdot \cdots \cdot \mathrm{E}\{\rho^{X_{n-1}}\} \\ &= \prod_{i=\tau}^{n-1} G_{\sigma,X_i}(\rho) \qquad (4) \end{aligned}$$

for the PGF of Y_τ, i.e. the PGF of Y_τ is the product of the PGFs of the $X_\tau, \ldots, X_{n-1}$ and thereby known. Note that the expectation of the product in (3) can be written as a product

of expectations since the channel is memoryless and thus the X_i are independent. The product (4) results directly from the definition (2) of the G_{σ,X_i}.

Using the PGF of Y_τ we can calculate the probability that there are ε, $0 \le \varepsilon \le n-\tau$, errors in $\mathbf{r}_\tau$ by

$$\Pr(Y_\tau = \varepsilon) := \left. \frac{G^{(\varepsilon)}_{\sigma,Y_\tau}(\rho)}{\varepsilon!} \right|_{\rho=0}, \tag{5}$$

where the superscript $^{(\varepsilon)}$ denotes the ε-th derivative.

Recall that the λ–decoder is capable of decoding ε errors and τ erasures if $\lambda\varepsilon + \tau \le d_{\min} - 1$. In case of τ, $0 \le \tau \le d_{\min}-1$, erasures the decoder will fail if the number of errors in the non–erased symbols is greater than $\frac{d_{\min}-1-\tau}{\lambda}$. Using (5), the probability of this event is determined by

$$\begin{aligned}\Pr\left(Y_\tau > \frac{d_{\min}-1-\tau}{\lambda}\right) &= 1 - \sum_{\varepsilon=0}^{\left\lfloor \frac{d_{\min}-1-\tau}{\lambda} \right\rfloor} \Pr(Y_\tau = \varepsilon) \\ &=: \mathrm{P}_\sigma(\tau). \end{aligned} \tag{6}$$

$\mathrm{P}_\sigma(\tau)$ is the residual codeword error probability as a function of the channel state σ and the number τ of erased symbols. Hence, the optimal choice of τ is

$$\tau^\star_\sigma := \arg \min_{0 \le \tau \le d_{\min}-1} \{\mathrm{P}_\sigma(\tau)\} \tag{7}$$

$$= \arg \max_{0 \le \tau \le d_{\min}-1} \left\{ \sum_{\varepsilon=0}^{\left\lfloor \frac{d_{\min}-1-\tau}{\lambda} \right\rfloor} \Pr(Y_\tau = \varepsilon) \right\}. \tag{8}$$

The residual codeword error probability is minimized by erasing the $\tau^\star_\sigma$ most unreliable symbols since from (7) we obtain

$$\mathrm{P}^\star_\sigma := \mathrm{P}_\sigma(\tau^\star_\sigma) = \min_{0 \le \tau \le d_{\min}-1} \{\mathrm{P}_\sigma(\tau)\},$$

which proves that adaptive erasing with $\tau^\star_\sigma$ as in (7) is at least as good as errors–only decoding with $\tau = 0$ and single–trial fixed erasing with some $\tau^\star_{\text{fixed}}$, $0 \le \tau^\star_{\text{fixed}} \le d_{\min}-1$ in terms of the achievable residual codeword error probability.

Using the results from this section we can state Algorithm 1 for optimal adaptive error/erasure decoding. It provides an exact solution for Problem 1.

The drawback of Algorithm 1 is its computational complexity. Sorting a vector of length n in line 2 has complexity $\mathcal{O}(n^2)$ and can be accomplished in place e.g. by the *bubble sort* algorithm [18]. Calculating the PGFs $G_{\sigma,Y_\tau}(\rho)$, $\tau = 0,\dots,d_{\min}-1$, in lines 4–5 essentially means multiplying n polynomials $G_{\sigma,X_i}(\rho)$, each with degree 1. This can be done efficiently using n *Fast Fourier Transforms (FFT)* of length n and componentwise multiplication of the frequency domain coefficients. Since the input polynomials for the FFT have degree 1 (e.g. only two non–zero coefficients), 2*–pruned* FFTs [19] with complexity $\mathcal{O}(n)$ can be used. The n 2–pruned FFTs together have complexity $\mathcal{O}(n^2)$ and the number of componentwise multiplications is n^2. The required single inverse FFT of length n has complexity $\mathcal{O}(n\log(n))$.

Algorithm 1: Optimal Adaptive Error/Erasure Decoding

input : $\mathcal{C}(2;n,k,d_{\min})$, $\mathbf{y} \in \mathbb{R}^n$, σ, λ–decoder $\mathrm{dec}_{\mathcal{C}}(\cdot)$

1 calculate $h_\sigma(y_0),\dots,h_\sigma(y_{n-1})$
2 sort $\mathbf{y}$ s.t. $h_\sigma(y_0) \ge \cdots \ge h_\sigma(y_{n-1})$ // $\mathcal{O}(n^2)$
3 $\mathbf{r} \leftarrow (\beta(\alpha(y_0)),\dots,\beta(\alpha(y_{n-1})))$
4 **for** $\tau = 0,\dots,d_{\min}-1$ **do** // $\mathcal{O}(n^2)$
5 calculate $G_{\sigma,Y_\tau}(\rho)$
6 $m \leftarrow 1$
7 **for** $\tau \leftarrow 0$ **to** $d_{\min}-1$ **do** // $\mathcal{O}(n^2 d_{\min})$
8 **for** $\varepsilon = 0,\dots,\frac{d_{\min}-1-\tau}{\lambda}$ **do** // $\mathcal{O}(n^2)$
9 calculate $G_{\sigma,Y_\tau}(\rho)^{(\varepsilon)}|_{\rho=0}$
10 **if** $\mathrm{P}_\sigma(\tau) < m$ **then** // $\mathcal{O}(n d_{\min})$
11 $\tau^\star_\sigma \leftarrow \tau$
12 $m \leftarrow \mathrm{P}_\sigma(\tau)$
13 calculate $\mathbf{r}_{\tau^\star_\sigma}$ from $\mathbf{r}$
14 revoke sorting of $\mathbf{r}_{\tau^\star_\sigma}$
15 **return** $\mathrm{dec}_{\mathcal{C}}\left(\mathbf{r}_{\tau^\star_\sigma}\right)$ // $\mathcal{O}(n^2)$

output: codeword estimate $\tilde{\mathbf{c}} \in \mathcal{C}$ or erasure $\mathbb{X}$

Hence, the complexity of lines 4–5 is $\mathcal{O}(n^2)$. The loop in lines 8–9 requires the evaluation of $\left\lfloor \frac{d_{\min}-1-\tau}{\lambda} \right\rfloor + 1$ derivatives at $\rho = 0$. This can be accomplished with complexity $\mathcal{O}\left(\left(\left\lfloor \frac{d_{\min}-1-\tau}{\lambda} \right\rfloor + 1\right) n\right) \subseteq \mathcal{O}(n^2)$ using an algorithm from Pankiewiczs [20] which is based on *Horner's Scheme*. The resulting values are required for the calculation of the $\mathrm{P}_\sigma(\tau)$ in line 10 as in (6). For each $\mathrm{P}_\sigma(\tau)$, a sum over $\left\lfloor \frac{d_{\min}-1-\tau}{\lambda} \right\rfloor + 1$ probabilities $\Pr(Y_\tau = \varepsilon)$ has to be calculated. Using the pre–computed values from lines 8–9, this can be accomplished with complexity $\mathcal{O}(n d_{\min})$. Since the loop in lines 7–12 is executed $d_{\min}$ times, its complexity is $\mathcal{O}(n^2 d_{\min})$. The complexity for λ–decoding in line 15 is $\mathcal{O}(n^2)$. Altogether the computational complexity of Algorithm 1 is $\mathcal{O}(n^2 d_{\min}) \subseteq \mathcal{O}(n^3)$.

Section IV addresses a computationally more efficient version of the algorithm which uses very good approximations of the $\mathrm{P}_\sigma(\tau)$.

Example 1 We consider the BCH code $\mathcal{C}(2;127,36,31)$ with a traditional BMD error/erasure decoder, i.e. $\lambda = 2$. The symbols $\{-1,+1\}$ are transmitted over an AWGN channel. In this case, the unreliability of received symbol y is

$$h_\sigma(y) = h_{\sigma,\mathrm{AWGN}}(y) := \frac{1}{1+\exp\left(\frac{2y\alpha(y)}{\sigma^2}\right)}.$$

Throughout the paper $\exp(\cdot)$ and $\log(\cdot)$ have base e. We assume $\mathrm{SNR} = 0\,\mathrm{dB}$, and obtain $\sigma = \sqrt{\frac{1}{2}\cdot 10^{\frac{-\mathrm{SNR}}{10}}} = \sqrt{0.5}$. Figure 1 depicts the operation of the loop in lines 7–12 of Algorithm 1. For each $\tau = 0,\dots,30$ and $\varepsilon = 0,\dots,\frac{30-\tau}{2}$ the probabilities $\Pr(Y_\tau = \varepsilon)$ are calculated. Each $\Pr(Y_\tau = \varepsilon)$ is represented by one point in Figure 1. This allows to calculate the sums in the maximization term of (8). Each of the sums is the sum over one slice of the point surface in Figure 1 in ε–direction. The optimal choice of τ is

the slice whose sum is maximal, in case of the considered codeword/transmission/received vector the optimization yields $\tau^{\star}_{\sqrt{0.5}} = 4$.

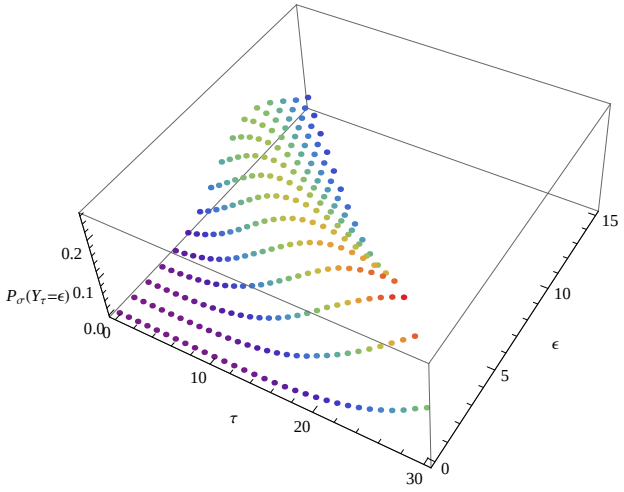

Fig. 1. Point surface consisting of the probabilities $\Pr(Y_\tau = \varepsilon)$, where $\tau = 0, \ldots, 30$ and $\varepsilon = 0, \ldots, \frac{30-\tau}{2}$.

IV. Computationally Efficient Adaptive Erasing

In this section, we present a technique which allows to reduce the computational complexity of Algorithm 1 from cubic in n to $\mathcal{O}(n^2 \sqrt[4]{n})$. It utilizes an approximation of the probabilities $\mathrm{P}_\sigma(\tau)$, $\tau = 0, \ldots, d_{\min} - 1$. This approximation is based on the following result by Hoeffding [21].

Theorem 1 (Hoeffding Bound) *Let $A_0, \ldots, A_{m-1}$ be m independent random variables with finite first and second moments, which are almost surely bounded, i.e.*

$$\Pr(A_i - E\{A_i\} \in [a_i, b_i]) = 1, \quad i = 0, \ldots, m-1,$$

where $E\{\cdot\}$ denotes the expectation of a random variable. Then, for the sum $S = A_0 + \cdots + A_{m-1}$ and $t > 0$ holds

$$\Pr(|S - E\{S\}| \geq mt) \leq 2 \exp\left(-\frac{2m^2 t^2}{\sum_{i=0}^{m-1} (b_i - a_i)^2} \right),$$

We apply Theorem 1 to $Y_\tau = \sum_{i=\tau}^{n-1} X_i$, i.e. $m = n-1-\tau$. By definition, we have $X_i \in \{0, 1\}$ and thus

$$\sum_{i=0}^{m-1} (b_i - a_i)^2 = m = n - 1 - \tau.$$

We obtain

$$\Pr\left(|Y_\tau - E\{Y_\tau\}| \geq t(n-1-\tau)\right) \leq 2 \exp\left(-2t^2(n-1-\tau)\right).$$

This means that the sum of the probabilities

$$\Pr(Y_\tau = 0), \ldots, \Pr(Y_\tau = E\{Y_\tau\} - t), \Pr(Y_\tau = E\{Y_\tau\} + t), \ldots, \Pr(Y_\tau = d_{\min} - 1) \quad (9)$$

is exponentially decreasing with t. We can conclude that the sum in (6) is dominated by only a small set of probabilities in proximity to the expectation $E\{Y_\tau\}$. Let us set $t := \frac{s}{n-1-\tau}$. We obtain

$$\Pr(|Y_\tau - E\{Y_\tau\}| \geq s) \leq 2 \exp\left(-\frac{2s^2}{\sqrt{n-1-\tau}} \right) \leq 2 \exp\left(-\frac{2s^2}{\sqrt{n}} \right),$$

i.e. the contribution of the probabilities from (9) in (6) is less than $2 \exp\left(-\frac{2s^2}{\sqrt{n}}\right)$. This fact can also be observed in Figure 1: The probabilities $\Pr(Y_\tau = \varepsilon)$ diminish quickly around the expectation of each slice in ε-direction. To obtain a good approximation (with precision goal 10^{-2}), let us select s such that

$$2 \exp\left(-\frac{2s^2}{\sqrt{n}} \right) < 10^{-2} \iff s > \sqrt{-\frac{\log(0.5 \cdot 10^{-2})}{2} \sqrt{n}}.$$

We define

$$s_0 := \begin{cases} \mathbb{N} \setminus \{0\} & \longrightarrow \quad \mathbb{N} \\ n & \longmapsto \quad \left\lfloor \sqrt{-\frac{\log(0.5 \cdot 10^{-2})}{2} \sqrt{n}} \right\rfloor + 1 \end{cases}.$$

Figure 2 shows the value of $s_0(n)$ for a practical range of code lengths n.

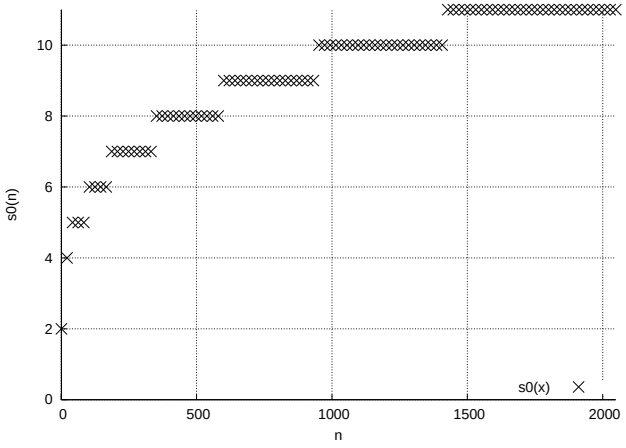

Fig. 2. Value of $s_0(n)$, $n = 1, \ldots, 2048$, for precision goal 10^{-2}.

Eventually, the Hoeffding bound justifies to neglect

$$\Pr(Y_\tau = 0), \ldots, \Pr(Y_\tau = E\{Y_\tau\} - s_0(n)), \Pr(Y_\tau = E\{Y_\tau\} + s_0(n)), \ldots, \Pr(Y_\tau = d_{\min} - 1)$$

in the sum of (6). As a result, we obtain very good approximations for $\mathrm{P}_\sigma(\tau)$ if we calculate the sum in (6) over at most $2s_0(n)$ elements, i.e.

$$\mathrm{P}_\sigma(\tau) \approx \widetilde{\mathrm{P}}_\sigma(\tau) := 1 - \sum_{\varepsilon = \max\{\lceil E\{Y_\tau\} \rceil - s_0(n),\, 0\}}^{\left\lfloor \min\left\{ E\{Y_\tau\} + s_0(n), \frac{d_{\min} - 1 - \tau}{\lambda} \right\} \right\rfloor} \Pr(Y_\tau = \varepsilon).$$

The required expectation can be easily calculated using the PGF (4) of Y_τ, i.e.

$$E\{Y_\tau\} := G_{\sigma,Y_\tau}(\rho)^{(1)}\Big|_{\rho=1}, \tag{10}$$

where the superscript $^{(1)}$ denotes the first derivative.

We use the previous results to state Algorithm 2 which solves Problem 1 with high precision and better computational complexity than Algorithm 1.

Algorithm 2: Efficient Adaptive Error/Erasure Decoding

input : $\mathcal{C}(2; n, k, d_{\min})$, $\mathbf{y} \in \mathbb{R}^n$, σ, $s_0(n)$, λ–decoder $\mathrm{dec}_{\mathcal{C}}(\cdot)$

1 calculate $h_\sigma(y_0), \ldots, h_\sigma(y_{n-1})$
2 sort $\mathbf{y}$ s.t. $h_\sigma(y_0) \geq \cdots \geq h_\sigma(y_{n-1})$ // $\mathcal{O}(n^2)$
3 $\mathbf{r} \leftarrow (\beta(\alpha(y_0)), \ldots, \beta(\alpha(y_{n-1})))$
4 **for** $\tau = 0, \ldots, d_{\min} - 1$ **do** // $\mathcal{O}(n^2)$
5 calculate $G_{\sigma,Y_\tau}(\rho)$
6 $m \leftarrow 1$
7 **for** $\tau \leftarrow 0$ **to** $d_{\min} - 1$ **do** // $\mathcal{O}(n\sqrt[4]{n}d_{\min})$
8 calculate $E\{Y_\tau\}$
9 $l \leftarrow \max\{\lceil E\{Y_\tau\}\rceil - s_0(n), 0\}$
10 $u \leftarrow \left\lfloor \min\{E\{Y_\tau\} + s_0(n), \frac{d_{\min}-1-\tau}{\lambda}\}\right\rfloor$
11 **for** $\varepsilon = l, \ldots, u$ **do** // $\mathcal{O}(n\sqrt[4]{n})$
12 calculate $G_{\sigma,Y_\tau}(\rho)^{(\varepsilon)}|_{\rho=0}$
13 **if** $\widetilde{\mathrm{P}}_\sigma(\tau) < m$ **then** // $\mathcal{O}(n\sqrt[4]{n})$
14 $\tau^\star_\sigma \leftarrow \tau$
15 $m \leftarrow \widetilde{\mathrm{P}}_\sigma(\tau)$
16 calculate $\mathbf{r}_{\tau^\star_\sigma}$ from $\mathbf{r}$
17 revoke sorting of $\mathbf{r}_{\tau^\star_\sigma}$
18 **return** $\mathrm{dec}_{\mathcal{C}}\left(\mathbf{r}_{\tau^\star_\sigma}\right)$ // $\mathcal{O}(n^2)$

output: codeword estimate $\tilde{\mathbf{c}} \in \mathcal{C}$ or erasure χ

Algorithm 2 has some differences compared to Algorithm 1, we will now briefly analyze their computational complexity.

Lines 1–6 remain unchanged, sorting, mapping to symbols of $\mathbb{F}_2$ and pre–calculation of the PGFs is the same for both the exact the the approximative algorithms. The loop in lines 7–15 starts with the calculation of the expectation $E\{Y_\tau\}$ according to (10). This can be accomplished with linear complexity. In lines 9–10, lower and upper bounds for the loop in lines 11–12 are calculated, using essentially $E\{Y_\tau\}$ and the input parameter $s_0(n)$. Since $s_0(n)$ grows with $\sqrt[4]{n}$, the loop in lines 11–12 calculates the value of $\mathcal{O}(\sqrt[4]{n})$ subsequent derivatives of the PGF $G_{\sigma,Y_\tau}(\rho)$. The complexity of this calculation is $\mathcal{O}(n\sqrt[4]{n})$ using Pankiewiczs' algorithm [20]. The calculation of $\widetilde{\mathrm{P}}_\sigma(\tau)$ in line 13 involves summation of $2\sqrt[4]{n}$ probabilities $\Pr(Y_\tau = \varepsilon)$. Using the pre–computed values of the derivatives from lines 11–12, each $\Pr(Y_\tau = \varepsilon)$ can be calculated with complexity linear in n, hence $\widetilde{\mathrm{P}}_\sigma(\tau)$ can be calculated with complexity $\mathcal{O}(n\sqrt[4]{n})$. Note that calculating $\mathrm{P}_\sigma(\tau)$ in Algorithm 1 is in $\mathcal{O}(nd_{\min})$. Alltogether, the loop in lines 7–15 is in $\mathcal{O}(n\sqrt[4]{n}d_{\min})$ and thus the overall complexity of Algorithm 2 is $\mathcal{O}(n^2\sqrt[4]{n})$.

V. Simulation Results

After the derivations of two adaptive error/erasure decoding algorithms in Sections III and IV, we devote this section to the analysis of their performance and behavior. First, we consider the short BCH code $\mathcal{C}(2; 31, 16, 7)$, a traditional BMD decoder with $\lambda = 2$ and an AWGN channel in the range between $0\,\mathrm{dB}$ and $6\,\mathrm{dB}$.

Figure 3 shows the simulation results. The black curve (diamonds) denotes traditional errors–only decoding. The green curve (squares) shows the result of Algorithm 1. It is not distinguishable from the red curve (circles) showing the result of the computationally more efficient Algorithm 2. For reference, the figure also contains the result of error/erasure decoding with fixed erasing (blue curve, triangles) as in [1]. The aforementioned result assumes very good channel conditions, hence its performance is bad in the considered range. However, there is a crossing point with the errors–only curve and we showed that the gain of optimal fixed erasing is $1.5\,\mathrm{dB}$ for an infinitively good channel. Note that the simulation confirms our observation from Section III, that Algorithm 1 must be as least as good as errors–only decoding **and** error/erasure decoding with optimal fixed erasing.

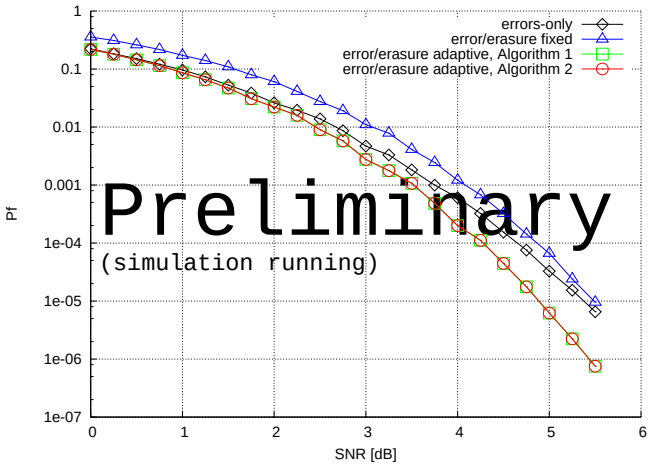

Fig. 3. Simulation results for $\mathcal{C}(2; 31, 16, 7)$.

For the second simulation, we reconsider the setting of Example 1, i.e. the BCH code $\mathcal{C}(2; 127, 36, 31)$. We observe that Algorithm 2 enables a reduction of the residual codeword error probability starting at around $\mathrm{SNR} = 1\,\mathrm{dB}$.

VI. Conclusions

Despite the seminal results of Kötter and Vardy about algebraic soft–decision decoding [22] using the *Guruswami–Sudan algorithm* [23], pseudo–soft decoding with traditional algebraic error/erasure decoders is still of practical interest. Such decoders are widely deployed and efficient implementations are available. Single- and multi–trial error/erasure decoding builds up on these decoders, i.e. they are provided

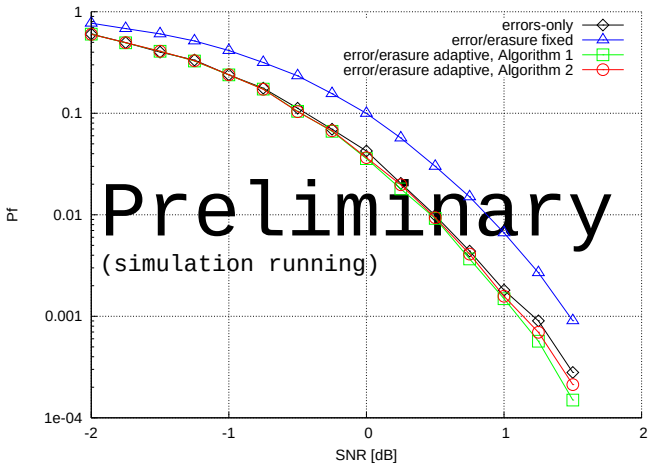

Fig. 4. Simulation results for $\mathcal{C}(2; 127, 36, 31)$.

with modified received vectors in which one or multiple sets of most unreliable symbols are erased.

In this paper, we provided two algorithms for adaptive single–trial error/erasure decoding for binary codes. The erasing strategy of the first algorithm is guaranteed to be optimal. The prize for this optimality is computational complexity $\mathcal{O}(n^3)$. The second algorithm gives an approximative optimal solution with precision 10^{-2}. This allows to reduce complexity to $\mathcal{O}(n^2 \sqrt[4]{n})$. Our simulations show that the performance results of both algorithms are virtually indistinguishable in practical settings. However, the approximative algorithm can easily be adapted to fulfill higher precision requirements.

Since our proposed algorithms are optimal, their residual codeword error probability is guaranteed to be superior compared to errors–only decoding and single–trial error/erasure decoding with an optimal fixed erasing strategy. It would be interesting to have an upper bound which proves the gain of adaptive erasing over errors–only and fixed single–trial error/erasure decoding. This bound is in focus of our current investigations.

Our work on the subject is continued with a generalization to multiple decoding trials and non–binary channels. This will enable our algorithms to be applied in existing coding standards which are based on serially concatenated coding schemes.

Acknowledgments

The authors would like to thank Serpil Senger and Alexander Zeh for carefully proofreading the manuscript.

References

[1] C. Senger, V. R. Sidorenko, and V. V. Zyablov, "On Generalized Minimum Distance decoding thresholds for the AWGN channel," in *Proc. XII Symposium Problems of Redundancy in Information and Control Systems*, St. Petersburg, Russia, May 2009, pp. 155–163. [Online]. Available: http://k36.org/redundancy2009/proceedings.pdf

[2] G. D. Forney, "Generalized Minimum Distance decoding," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 125–131, April 1966.

[3] ——, *Concatenated Codes*. Cambridge, MA, USA: M.I.T. Press, 1966.

[4] C. Senger, V. R. Sidorenko, M. Bossert, and V. V. Zyablov, "Decoding generalized concatenated codes using interleaved Reed–Solomon codes," in *Proc. IEEE Int. Symposium on Inform. Theory*, Toronto, ON, Canada, July 2008. [Online]. Available: http://dx.doi.org/10.1109/ISIT.2008.4595300

[5] ——, "Multi-trial decoding of concatenated codes using fixed thresholds," *Problems of Information Transmission*, vol. 46, no. 2, 2010.

[6] V. R. Sidorenko, C. Senger, M. Bossert, and V. V. Zyablov, "Single–trial decoding of concatenated codes using fixed or adaptive erasing," *Advances in Mathematics of Communications (AMC)*, vol. 4, no. 1, pp. 49–60, February 2010. [Online]. Available: http://dx.doi.org/10.3934/amc.2010.4.49

[7] C. Senger, V. R. Sidorenko, M. Bossert, and V. V. Zyablov, "Optimal thresholds for GMD decoding with $\frac{\ell+1}{\ell}$–extended bounded distance decoders," in *Proc. IEEE Int. Symposium on Inform. Theory*, Austin, TX, USA, June 2010. [Online]. Available: http://arxiv.org/abs/1001.1106

[8] E. L. Blokh and V. V. Zyablov, *Linear Concatenated Codes*. Nauka, 1982, in Russian.

[9] J. H. Weber and K. A. S. Abdel-Ghaffar, "Reduced GMD decoding," *IEEE Trans. Inform. Theory*, vol. IT-49, no. 4, pp. 1013–1027, April 2003. [Online]. Available: http://dx.doi.org/10.1109/TIT.2003.809504

[10] S. I. Kovalev, "Two classes of minimum generalized distance decoding algorithms," *Problems of Information Transmission*, vol. 22, no. 3, pp. 186–192, 1986, translated from Russian, original in Problemy Peredachi Informatsii, pp. 35–42.

[11] V. R. Sidorenko, C. Senger, M. Bossert, and V. V. Zyablov, "Single-trial adaptive decoding of concatenated codes," in *Proc. International Workshop on Algebraic and Combinatorial Coding Theory*, Pamporovo, Bulgaria, June 2008. [Online]. Available: http://www.moi.math.bas.bg/acct2008/b44.pdf

[12] V. R. Sidorenko, A. Chaaban, C. Senger, and M. Bossert, "On extended Forney–Kovalev GMD decoding," in *Proc. IEEE Int. Symposium on Inform. Theory*, Seoul, Korea, July 2009. [Online]. Available: http://dx.doi.org/10.1109/ISIT.2009.5205900

[13] A. Chaaban, V. R. Sidorenko, and C. Senger, "On multi–trial Forney–Kovalev decoding of concatenated codes," 2010, preprint.

[14] G. D. Forney, "On decoding BCH codes," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 549–557, October 1965.

[15] R. E. Blahut, *Algebraic Codes for Data Transmission*, 1st ed. Cambridge: Cambridge University Press, 2003, ISBN 0-521-55374-1.

[16] W. Feller, *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition*. John Wiley & Sons, January 1968, ISBN 0-471-25708-7.

[17] J. A. Gubner, *Probability and Random Processes for Electrical and Computer Engineers*. New York, NY, USA: Cambridge University Press, 2006, ISBN 0-52186-470-4.

[18] D. E. Knuth, *Sorting and Searching*, 2nd ed., ser. The Art of Computer Programming. Addison-Wesley, 1998, ISBN 0-201-89685-0.

[19] H. V. Sorensen and C. S. Burrus, "Efficient computation of the DFT with only a subset of input or output points," *IEEE Trans. Signal Processing*, vol. 41, no. 3, pp. 1184–1200, 1993. [Online]. Available: http://dx.doi.org/10.1109/78.205723

[20] W. Pankiewicz, "Algorithms: Algorithm 337: calculation of a polynomial and its derivative values by horner scheme," *Commun. ACM*, vol. 11, no. 9, p. 633, 1968. [Online]. Available: http://doi.acm.org/10.1145/364063.364089

[21] W. Hoeffding, "Probability inequalities for sums of bounded random variables." *J. Amer. Statist. Assoc.*, vol. 58, pp. 13—30, 1963.

[22] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed–Solomon codes," *IEEE Trans. Inform. Theory*, vol. IT-49, no. 11, pp. 2809–2825, November 2003. [Online]. Available: http://dx.doi.org/10.1109/TIT.2003.819332

[23] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. IT-45, no. 6, pp. 1755–1764, September 1999. [Online]. Available: http://dx.doi.org/10.1109/18.782097